UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/661,167	09/12/2003	Dmitry M. Smirnov	021756-017900US	7522

51206   7590   03/23/2010
TOWNSEND AND TOWNSEND AND CREW LLP/ORACLE
TWO EMBARCADERO CENTER
8TH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER
GOFMAN, ALEX N

ART UNIT	PAPER NUMBER
2162	

MAIL DATE	DELIVERY MODE
03/23/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/661,167	SMIRNOV ET AL.
	Examiner	Art Unit	
	ALEX GOFMAN	2162	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *14 December 2009*.
2a) ☒ This action is **FINAL**.   2b) ☐ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1,2,44-48 and 50-68* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) *1,2,44-48 and 50-68* is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.
10) ☒ The drawing(s) filed on *12 September 2003* is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All   b) ☐ Some * c) ☐ None of:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____.

      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Amendment submitted December 14, 2009 has been considered by examiner.
Claims 1-2, 44-48 and 50-68 are pending.

Response to Arguments

1.   Applicant's arguments with respect to amended Claims 1, 7, 14, 20, 32, 44,

57, 59, 62 and 65 have been considered but do not overcome the cited prior art.

Specific arguments are addressed below.

        As to arguments of Claim 1, the Applicant states that Lee does not

disclose the newly added limitation "**generate an output data file** giving a data

structure responsive to a data structure of the selected first and second

compound transform definitions." The Examiner respectfully disagrees.

        Lee [0293] states "In one embodiment of the present invention, the Output

XML is formed using a XML template obtained during the translation process

(step 1606) and display characteristics." Lee [0329] further states "Server 40

combines the Output XML with its associated XSL stylesheets (step 1800).

Identity Server 40 then formats the resulting combination as an HTML output

document (step 1802)."

The result of the transformation is "Output XML." The "Output XML" is then shown via an "HTML output document. The HTML output is the output data file as required by the above argued claim limitation.

The Applicant also states that there is no motivation to combine Lee and Chung. The Examiner respectfully disagrees.

Lee discloses the transformation of data and the generation of the output. Lee, however, does not disclose the limitation of "concurrently navigate the selected first compound transform definition, the selected second compound transform definition, and the data to be transformed using the parallel processing thread, navigation within the data to be transformed being responsive to transform definitions within the selected first and second transform definition." The reason for using Chung is in order to cure the deficiency of Lee. Furthermore, the Examiner has provided the Graham v Deere factual inquiries below:

A. Determining the Scope and Content of the Prior Art

Both Lee and Chung deal with transformation of data and are thus in the same "scope and content of the prior art."

B. Ascertaining the Differences Between the Claimed Invention and the Prior Art

The Claimed invention describes the transformation of at least two types of data where the transformation of the at least two types of data are performed concurrently.

Lee describes data transformation, while Chung describes concurrent transformation of data.

C. Resolving the Level of Ordinary Skill in the Art

The reasoning for "the Level of Ordinary Skill in the Art" of why Chung was brought in, may be found in Chung (Col 7 In 21-22), "Thereby data contention is reduced and throughput is increased."

Therefore, based on the Graham v Deere factual inquires described above, Chung was appropriately used to combine with Lee.

Claim Rejections - 35 USC § 103

2.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3.   Claims 1-42, 44-48 and 50-68 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee (US Patent Application Publication 2002/0129024),

hereinafter, <u>Lee</u> in view of <u>Chung et al</u> (US Patent 6,850,947), hereinafter,

<u>Chung</u>.

     Claim 1: <u>Lee</u> discloses a system comprising:

     a. a computing device; a data interface on the computing device

configured to receive data to be transformed or to send transformed data [0323].

     b. memory configured to store one or more transform process definitions

having at least one simple transform definition and at least one compound

transform definition [0293, 0305, 0312]. [The "transform definition" is the format

into which the incoming data will be formatted. As for simple and compound

definitions, see example of transformation of data from [0305] into example in

[0312].]

     c. an application including computer instructions [0293].

     d. a data interpreter configured to exchange data with the data interface

and the application, the data interpreter including a transform engine configured

to

     d1. select a first compound transform definition from the one or more

transform process definitions, the selected first compound transform definition

including a hierarchical data structure with a first sub-definition [0305, 0312,

0323].

d2. invoke a first processing thread to process the first compound definition including the first sub-definition [0305, 0307, 0312].

d3. select a second compound transform definition from the one or more transform process definitions, the selected second compound transform definition including a hierarchical data structure with a second sub-definition [0305, 0307, 0312].

d4. invoke a second processing thread to process the second compound transform definition including the second sub-definition [0305, 0307, 0312].

Lee, as discussed above, discloses data transformation using a plurality of templates (templates are similar to transform definitions). However, Lee lacks the concurrent processing of multiple transform definitions. Specifically, Lee does not explicitly disclose:

d5. concurrently navigate the selected first compound transform definition, the selected second compound transform definition, and the data to be transformed using the parallel processing thread, navigation within the data to be transformed being responsive to transform definitions within the selected first and second transform definition.

However, Chung does (Col 7 ln 15-22; Col 7 ln 52-66). Chung discloses concurrently transforming data with multiple transformation definitions. The transform definitions are for example, aggregation transformation, filter transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify <u>Lee</u> with <u>Chung</u>. One would have been motivated to do so in order to concurrently transform data so that "data contention is reduced and throughput is increased."

<u>Lee</u> as modified further discloses:

d6. generate an output file data having a data structure responsive to a data structure of the selected first and second compound transform definitions [0293, 0308, 0329]. [The result of the transformation is "Output XML." The "Output XML" is then shown via an "HTML output document. The HTML output is the output data file.]

Claim 2: <u>Lee</u> as modified discloses the system of Claim 1 above, and further discloses wherein the transformation engine is further configured to process the at least one compound transform definition using recursion [0247, 0305, 0312]. [Recursion is performed on the nested (subflow) elements.]

Claim 3: <u>Lee</u> as modified discloses the system of Claim 1 above, and further discloses wherein the data interpreter is further configured to support a plurality of applications [0307].

Claim 4: <u>Lee</u> as modified discloses the system of Claim 1 above, and further discloses wherein the application is a database application, accounting application, human resources application, customer management application, inventory application, or an internet application [0307].

Claim 5: <u>Lee</u> as modified discloses the system of Claim 1 above, and further discloses wherein the application and the data interpreter are integrated [0307, 0308].

Claim 6: <u>Lee</u> as modified discloses the system of Claim 1 above, and further discloses the data interpreter further includes a computing device configured to support the transform engine [0118].

Claim 7: <u>Lee</u> discloses a data interpreter configured to transform data to be transformed, the data interpreter comprising:

a. at least one computing device [0118].

b. a transform engine supported by the computing device, the transform engine being configured to

b1. access a transform process definition including a hierarchical data structure of transform definitions, the data structure including a simple transform definition, a first compound transform definition having a first sub-definition and a second compound transform definition having a second sub-definition [0293, 0305, 0312].

b2. invoke a first processing thread to process the first compound definition including the first sub-definition [0305, 0307, 0312].

b3. invoke a second processing thread to process the second compound transform definition including the second sub-definition [0305, 0307, 0312].

Lee, as discussed above, discloses data transformation using a plurality of templates (templates are similar to transform definitions). However, Lee lacks the concurrent processing of multiple transform definitions. Specifically, Lee does not explicitly disclose:

b4. concurrently navigate the selected first compound transform definition, the selected second compound transform definition, and the data to be transformed using the parallel processing thread, navigation within the data to be transformed being responsive to transform definitions within the selected first and second transform definition.

However, Chung does (Col 7 ln 15-22; Col 7 ln 52-66). Chung discloses concurrently transforming data with multiple transformation definitions. The transform definitions are for example, aggregation transformation, filter transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify Lee with Chung. One would have been motivated to do so in order to concurrently transform data so that "data contention is reduced and throughput is increased."

Lee as modified further discloses:

b6. generate an output data file having a data structure responsive to a data structure of the selected first and second compound transform definitions [0308, 0329].

Claim 8: <u>Lee</u> as modified discloses the apparatus of Claim 7 above and further discloses wherein the data structure of the output data file is responsive to a structure of transform process definition [0312].

Claim 9: <u>Lee</u> as modified discloses the apparatus of Claim 7 above and further discloses wherein the transform engine is configured to process the compound transform definition using recursion [0247, 0305, 0312].

Claim 10: <u>Lee</u> as modified discloses the apparatus of Claim 7 above and further discloses wherein the transform engine is configured to generate output data including data elements characterized by the transform process definition and having no contribution from the data to be transformed [0308]. [If the transformation template does not contain an output for a specific element, it will not "contribute" that element to the output.].

Claim 11: <u>Lee</u> as modified discloses the apparatus of Claim 7 above and further discloses wherein the transform process definition is configured such that some data elements in the data to be transformed do not make a contribution to the output data [0308]. [If the transformation template does not contain an output for a specific element, it will not "contribute" that element to the output.].

Claim 12: <u>Lee</u> as modified discloses the apparatus of Claim 7 above and further discloses wherein the transform engine is further configured to navigate the data to be transformed responsive to the data structure of the transform definitions within the transform process definition [0247, 0305, 0312].

Claim 13: <u>Lee</u> as modified discloses the apparatus of Claim 7 above and further discloses wherein the transform engine is further configured to navigate the data to be transformed responsive to content of the transform definitions [0247, 0305, 0312].

Claim 14: <u>Lee</u> discloses a method of transforming data using an application programming interface, the method comprising:

a. receiving data to be transformed at the application programming interface [0116, 0304].

b. parsing identification data within the data to be transformed, the identification data characterizing the data to be transformed [0304].

c. using the identification data to select a first compound transform process definition from a set of one or more transform process definitions, the selected first compound transform definition having a first sub-definition and defining a process of translating data elements within the data to be transformed to output data elements [0304, 0305, 0312].

d. invoking a first processing thread to process the first compound definition including the first sub-definition [0305, 0307, 0312].

e. using the identification data to select a second compound transform definition from the set of one or more transform process definitions, the selected second compound transform definition having a second sub-definition and

defining a process of translating data elements within the data to be transformed

to output data elements [0305, 0307, 0312].

f. invoking a second processing thread to process the second compound

transform definition including the second sub-definition [0305, 0307, 0312].

g. transforming the data to be transformed to an output data file, using a

transformation engine and the selected first and second compound transform

definitions, a data structure of the output data being responsive to a data

structure of the first and second compound transform definitions [0305, 0312,

0329].

Lee, as discussed above, discloses data transformation using a plurality of

templates (templates are similar to transform definitions). However, Lee lacks the

concurrent processing of multiple transform definitions. Specifically, Lee does not

explicitly disclose:

However, Chung does (Col 7 ln 15-22; Col 7 ln 52-66). Chung discloses

concurrently transforming data with multiple transformation definitions. The

transform definitions are for example, aggregation transformation, filter

transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at

the time the invention was made to modify Lee with Chung. One would have

been motivated to do so in order to concurrently transform data so that "data

contention is reduced and throughput is increased."

Claim15: <u>Lee</u> as modified discloses the method of Claim 14 above and further discloses wherein transforming the data to be transformed includes nesting of data records [0305, 0312].

Claim 16: <u>Lee</u> as modified discloses the method of Claim 14 above and further discloses wherein transforming the data to be transformed includes calling a transformation process recursively responsive to a compound transform definition in the selected transform process definition [0247, 0305, 0312].

Claim 17: <u>Lee</u> as modified discloses the method of Claim 14 above and further discloses wherein the selected first and second transform definitions are selected based on information within the identification data that identifies a destination of the data to be transformed [0304, 0305, 0312].

Claim 18: <u>Lee</u> as modified discloses the method of Claim 14 above and further discloses wherein the first and second transform definitions are selected based on information within the identification data that identifies a format of the output data [0304, 0305, 0312].

Claim 19: <u>Lee</u> as modified discloses the method of Claim 14 above and further discloses wherein the transform process definition includes a extensible markup language (XML) [0323].

Claim 20: <u>Lee</u> discloses a method of transforming data using an application programming interface, the method comprising:

    a. receiving data to be transformed at the application programming interface, the data to be transformed including identification data [0304].

    b. using the identification data to select a first compound transform definition from a set of transform process definitions, the selected first compound transform definition having a first sub-definition and defining a process of translating data elements within data to be transformed to output data elements [0304, 0305, 0312, 0323].

    c. invoking a first processing thread to process the first compound definition including the first sub-definition [0305, 0307, 0312].

    e. using the identification data to select a second compound transform definition from the set of one or more transform process definitions, the selected second compound transform definition having a second sub-definition and defining a process of translating data elements within the data to be transformed to output data elements [0305, 0307, 0312].

    f. invoking a second processing thread to process the second compound transform definition including the second sub-definition [0305, 0307, 0312].

    Lee, as discussed above, discloses data transformation using a plurality of templates and transforming the data to be transformed to an output data file [0293, 0329] (templates are similar to transform definitions). However, Lee lacks the concurrent processing of multiple transform definitions. Specifically, Lee does not explicitly disclose:

g. concurrently navigating the data to be transformed, the first compound

transform definition, and the second compound transform definition using the

parallel processing threads navigation in the data to be transformed being

responsive to the transform process definition.

However, Chung does (Col 7 ln 15-22; Col 7 ln 52-66). Chung discloses

concurrently transforming data with multiple transformation definitions. The

transform definitions are for example, aggregation transformation, filter

transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at

the time the invention was made to modify Lee with Chung. One would have

been motivated to do so in order to concurrently transform data so that "data

contention is reduced and throughput is increased."

Claim 21: Lee as modified discloses the method of Claim 20 above and

further discloses wherein the selected first and second compound transform

definitions are selected based on information within the identification data that

identifies a format of the data to be transformed [0304, 0305, 0312].

Claim 22: Lee as modified discloses the method of Claim 20 above and

further discloses wherein the selected first and second compound transform

definitions are selected based on information within the identification data that

identifies a source of the data to be transformed [0304, 0305, 0312].

Claim 23: Lee as modified discloses the method of Claim 20 above and further discloses wherein the application programming interface is shared by several applications [0118].

Claim 24: Lee as modified discloses the method of Claim 20 above and further discloses wherein the application programming interface is shared by several applications and the selected first and second compound transform definitions are selected based on an identity of one of the several applications [0304].

Claim 25: Lee as modified discloses the method of Claim 20 above and further discloses wherein a structure of the output data file is responsive to a structure of a transform definition included in the selected first and second compound transform definitions [0304, 0305,0312].

Claim 26: Lee as modified discloses the method of Claim 20 above and further discloses wherein transforming the data to be transformed includes nesting of data records [0305, 0312].

Claim 27: Lee as modified discloses the method of Claim 20 above and further discloses wherein transforming the data to be transformed includes filtering of data records [0305, 0312].

Claim 28: Lee as modified discloses the method of Claim 20 above and further discloses wherein transforming the data to be transformed includes calling a transformation process recursively responsive to a data structure of a transform

definition included in the selected first and second compound transform

definitions [0247, 0305, 0312].

Claim 29: Lee as modified discloses the method of Claim 20 above and

further discloses wherein transforming the data to be transformed includes

searching the data to be transformed for a data field specified in a transform

definition included in the transform process definition [0308].

Claim 30: Lee as modified discloses the method of Claim 20 above and

further discloses wherein the transform definition includes a translation codeset

parameter configured to invoke an external reference [0307].

Claim 31: Lee as modified discloses the method of Claim 20 above and

further discloses wherein the transform definition includes a translation codeset

parameter configured to invoke an external reference, the external reference

being configured to perform logic operations using the data to be transformed

[0307].

Claim 32: Lee discloses a method of transforming data, the method

comprising:

a. positioning a definition pointer to point at a first compound transform

definition within a transform process definition [0321]. [The program identifies

entries for a name, Bob Smith. The name may serve as a specific identifier like a

pointer. Information related to Bob Smith is found and transformed into the

appropriate format.]

b. invoking a processing thread to read the pointed at first compound

transform definition [0321].

c. searching data to be transformed for a data element to be transformed,

the search being responsive to the first compound transform definition [0321].

d. transforming any found data element into an output data file, responsive

to the first compound transform definition, a data structure of the output data file

being responsive to a data structure of the transform process definition [0321,

0323, 0329].

e. positioning a definition pointer to point at a second compound transform

definition within the transform process definition [0321].

f. invoking a second processing thread to read the pointed at second

compound transform definition [0321, 0323]. [*The pointing to a specific "process

definition" is done once a specific template is identified.*]

g. searching data to be transformed for another data element to be

transformed, the search being responsive to the second compound transform

definition [0321, 0323].

h. transforming any found data element into the output data file, responsive

to the second compound transform definition, the data structure of the output

data file being responsive to the data structure of the transform process definition

[0308

i. wherein the read first compound transform definition includes a translation codeset parameter enabling the transforming to include a call to one of a function or a lookup table located in the first compound transform definition (Figure 37, [0302]). [The data registry file is the "codeset parameter" since it is a lookup table which contains data used for determining a specific output structure.]

Lee, as discussed above, discloses data transformation using a plurality of templates (templates are similar to transform definitions). However, Lee lacks the concurrent/parallel processing of multiple transform definitions.

However, Chung does (Col 7 ln 15-22; Col 7 ln 52-66). Chung discloses concurrently transforming data with multiple transformation definitions. The transform definitions are for example, aggregation transformation, filter transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify Lee with Chung. One would have been motivated to do so in order to concurrently transform data so that "data contention is reduced and throughput is increased."

Claim 33: Lee as modified discloses the method of Claim 32 above and further discloses determining a type of the read first compound transform definition and, if the first compound transform definition is not a simple transform definition type, recursively calling the method of claim [0247, 0305, 0312].

Claim 34: <u>Lee</u> as modified discloses the method of Claim 32 above and further discloses including determining if all sub-definitions of a compound transform definition have been processed [0293, 0305, 0312].

Claim 35: <u>Lee</u> as modified discloses the method of Claim 32 above and further discloses wherein the method of transforming data includes nesting of a data element [0305, 0312].

Claim 36: <u>Lee</u> as modified discloses the method of Claim 32 above and further discloses if no data element is found in the step of searching data to be transformed, adding an output data file element to the output data responsive to the read first compound transform definition, the data to be transformed having no contribution to the output data element [0308, 0329].

Claim 37: <u>Lee</u> as modified discloses the method of Claim 32 above and further discloses wherein the read transform definition includes a value parameter configured to specify a value for inclusion in the output data file [0318].

Claim 38: <u>Lee</u> as modified discloses the method of Claim 32 above and further discloses wherein the data element is a compound data element and the read transform definition includes a source record parameter configured to specify the compound data element [0308].

Claim 39: <u>Lee</u> as modified discloses the method of Claim 32 above and

further discloses wherein the read first compound transform definition is in a

meta-language format [0308, 0312].

Claim 40: <u>Lee</u> as modified discloses the method of Claim 32 above and

further discloses wherein the data to be transformed data is in a meta-language

data format [0308].

Claim 41: <u>Lee</u> as modified discloses the method of Claim 32 above and

further discloses wherein the read first compound transform definition includes a

transform element having an output field name and a source field parameter

[0305, 0308, 0312].

Claim 42: <u>Lee</u> as modified discloses the method of Claim 32 above and

further discloses wherein the read first compound transform definition includes a

value parameter configured to populate a field in the output data file [0305, 0308,

0312].

Claim 44: <u>Lee</u> discloses a method of transforming data, the method

comprising:

a. positioning a definition pointer to point at a transform definition, the first

compound transform definition being one of a plurality of transform definitions

within a transform process definition [0321].

b. invoking a first processing thread to read the pointed at first compound transform definition and sub-definitions of the first compound transform definition [0321].

c. positioning a first payload pointer to point at a data element to be transformed, the positioning being responsive to a data structure of the first compound transform definition [0321].

d. transforming the data element into an output data file, responsive to the read first compound transform definition [0321, 0329].

e. positioning the definition pointer to point at a second compound transform definition, the second compound transform definition being within the transform process definition [0321].

f. invoking a second processing thread to read the pointed at second compound transform definition and sub-definitions of the second compound transform definition [0321, 0323].

g. positioning a second payload pointer to point at a data element to be transformed, the positioning being responsive to a data structure of the second compound transform definition [0321, 0323].

h. transforming the second data element into the output data file, responsive to the read second compound transform definition [0308].

Lee, as discussed above, discloses data transformation using a plurality of templates (templates are similar to transform definitions). However, Lee lacks the concurrent/parallel processing of multiple transform definitions.

However, Chung does (Col 7 ln 15-22; Col 7 ln 52-66). Chung discloses concurrently transforming data with multiple transformation definitions. The transform definitions are for example, aggregation transformation, filter transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify Lee with Chung. One would have been motivated to do so in order to concurrently transform data so that "data contention is reduced and throughput is increased."

Claim 45: Lee as modified discloses the method of Claim 44 above and further discloses determining a type of the read first compound transform definition and, if the read first compound transform definition is not a simple transform definition type, recursively calling the method of claim 44 [0247, 0305, 0312].

Claim 46: Lee as modified discloses the method of Claim 44 above and further discloses determining a type of the read first compound transform definition and, if the read first compound transform definition is not a simple transform definition type, recursively calling the method of claim 44, wherein the

recursive call is responsive to the data structure of the transform process

definition [0305, 0308, 0312].

Claim 47: Lee as modified discloses the method of Claim 44 above and

further discloses further including determining a type of the read first compound

transform definition, if the read transform definition is not a simple transform

definition type recursively calling the method of claim 44, and determining if all

sub-elements of a compound element have been transformed [0305, 0308,

0312].

Claim 48: Lee as modified discloses the method of Claim 44 above and

further discloses further including determining if all sub-elements of a compound

element have been transformed and, if the determination returns a value of YES,

returning to a calling process [0305, 0308, 0312].

Claim 50: Lee as modified discloses the method of Claim 44 above and

further discloses wherein the method of transforming data includes un-nesting of

the data element to be transformed [0305, 0308, 0312].

Claim 51: Lee as modified discloses the method of Claim 44 above and

further discloses wherein the read first compound transform definition includes a

source field parameter configured to specify the data element [0308].

Claim 52: Lee as modified discloses the method of Claim 44 above and

further discloses wherein the read first compound transform definition includes a

source record parameter configured to specify the compound data element

[0305, 0308, 0312].

Claim 53: <u>Lee</u> as modified discloses the method of Claim 44 above and

further discloses wherein the read first compound transform definition includes a

translation codeset configured for calling computer instructions including logic

operations [0305, 0308, 0312].

Claim 54: <u>Lee</u> as modified discloses the method of Claim 53 above and

further discloses wherein the computer instructions are configured to call an

external process [0307].

Claim 55: <u>Lee</u> as modified discloses the method of Claim 53 above and

further discloses including a step of combining the data element with the

transform process definition prior to transforming the data element to output data

[0308].

Claim 56: <u>Lee</u> as modified discloses the method of Claim 53 above and

further discloses wherein the transform process definition includes a tree data

structure [0139].

Claim 57: <u>Lee</u> discloses a method of preparing data for transformation, the

method comprising:

a. receiving data to be transformed [0307].

b. parsing the received data to determine identification information [0307].

c. invoking a processing thread to use the identification information to extract a first compound transform definition from a plurality of transform process definitions, the extracted first compound transform definition including a transform definition configured to transform the data to be transformed, to direct navigation within the data to be transformed during transformation, and to determine a data structure of an output data file resulting from transformation of the data to be transformed, the first compound transform definition being configured to generate a compound data element in the output data file [0305, 0307, 0312].

d. invoking a processing thread to use the identification information to extract a second compound transform definition from the plurality of transform process definitions, the extracted second compound transform definition including a transform definition configured to transform the data to be transformed, to direct navigation within the data to be transformed during transformation, and to determine the data structure of the output data file resulting from transformation of the data to be transformed, the second compound transform definition being configured to generate a compound data element in the output data file [0305, 0307, 0312].

Lee, as discussed above, discloses data transformation using a plurality of templates (templates are similar to transform definitions). However, Lee lacks the concurrent/parallel processing of multiple transform definitions.

However, <u>Chung</u> does (Col 7 ln 15-22; Col 7 ln 52-66). <u>Chung</u> discloses

concurrently transforming data with multiple transformation definitions. The

transform definitions are for example, aggregation transformation, filter

transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at

the time the invention was made to modify <u>Lee</u> with <u>Chung</u>. One would have

been motivated to do so in order to concurrently transform data so that "data

contention is reduced and throughput is increased."

e. <u>Lee</u> as modified further discloses adding the extracted transform

process definition to meta-language transform input data including the data to be

transformed [0305, 0307, 0312].

Claim 58: <u>Lee</u> as modified discloses the method of Claim 57 above, and

<u>Lee</u> further discloses wherein the extracted transform process definition is in a

meta-language format [0305, 0307, 0312].

Claim 59: <u>Lee</u> discloses a computer readable media having embodied

thereon data, the data comprising:

a. computer instructions configured to position a definition pointer to point

at a first compound transform definition, the first compound transform definition

being within a transform process definition [0305, 0307, 0312].

b. computer instructions configured to invoke a first processing thread to read the pointed at first compound transform definition and sub-definitions of the first compound transform definition [0305, 0307, 0312].

c. computer instructions configured to increment a first payload pointer, within the data to be transformed, to a data element to be transformed, the incrimination being responsive to the pointed at first compound transform definition [0305, 0307, 0312].

d. computer instructions configured to transform any found data element into an output data file, responsive to the first compound transform definition [0305, 0307, 0312, 0329].

e. computer instructions configured to increment a second payload pointer, within the data to be transformed, to a data element to be transformed, the incrementation being responsive to the pointed at second compound transform definition [0305, 0307, 0312].

f. computer instructions configured to invoke a second processing thread to read the pointed at second compound transform definition and sub-definitions of the second compound transform definition [0305, 0307, 0312].

g. computer instructions configured to transform the second data element into the output data file, responsive to the read second compound transform definition [0305, 0307, 0312].

Lee, as discussed above, discloses data transformation using a plurality of templates (templates are similar to transform definitions). However, Lee lacks the concurrent/parallel processing of multiple transform definitions.

However, Chung does (Col 7 ln 15-22; Col 7 ln 52-66). Chung discloses concurrently transforming data with multiple transformation definitions. The transform definitions are for example, aggregation transformation, filter transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify Lee with Chung. One would have been motivated to do so in order to concurrently transform data so that "data contention is reduced and throughput is increased."

Claim 60: Lee as modified discloses the media of Claim 59 above, and further discloses wherein the data further comprises computer instructions configured to employ recursion to transform a compound data element within the data to be transformed [0305, 0307, 0312].

Claim 61: Lee as modified discloses the media of Claim 59 above, and further discloses wherein the data further comprises computer instructions configured to transform the data to be transformed using parallel processes [0305, 0307, 0312].

Claim 62: Lee discloses a computer readable storage media having embodied thereon data, the data comprising:

a. payload data including data to be transformed, the data to be

transformed including metadata characterizing simple data elements and

compound data elements [0305, 0307, 0312].

b. a transform process definition including a first and second compound

transform definition configured to transform the data to be transformed, to direct

navigation within the data to be transformed during transformation, and to

determine a data structure of an output data file resulting from the transformation,

the first compound transform definition also including a pointer to a function

enabled to perform logical operations on the data to be transformed during

transformation and generate the output data, the first compound transform

definition being configured to generate a compound data element in the output

data file using processing threads configured to navigate the first and second

compound transform definitions [0305, 0307, 0312].

Lee, as discussed above, discloses data transformation using a plurality of

templates (templates are similar to transform definitions). However, Lee lacks the

concurrent/parallel processing of multiple transform definitions.

However, Chung does (Col 7 ln 15-22; Col 7 ln 52-66). Chung discloses

concurrently transforming data with multiple transformation definitions. The

transform definitions are for example, aggregation transformation, filter

transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify <u>Lee</u> with <u>Chung</u>. One would have been motivated to do so in order to concurrently transform data so that "data contention is reduced and throughput is increased."

Claim 63: <u>Lee</u> as modified discloses the media of Claim 62 above, and further discloses wherein the computer readable media includes memory included in a data interface [0118].

Claim 64: <u>Lee</u> as modified discloses the media of Claim 62 above, and further discloses wherein the computer readable media includes a hard drive [0118].

Claim 65: <u>Lee</u> discloses an application system comprising:

a. means for positioning a definition pointer to point at a first compound transform definition within a transform process definition [0321].

b. means for invoking a first processing thread to read the first compound transform definition by the computing device [0321].

c. means for positioning the definition pointer to a point at a second compound transform definition within the transform process definition [0305, 0307, 0312].

d. means for invoking a second processing thread to read the second compound transform definition by the computing device [0305, 0307, 0312].

e. means for positioning a payload pointer to point to a first data element, the first data element being a member of a plurality of data elements within data to be transformed [0305, 0307, 0312].

f. means for generating an output data file using the first data element and the first and second compound transform definitions [0305, 0307, 0312, 0329].

g. wherein the means for positioning the definition pointer can be invoked concurrently with the means for positioning the payload pointer [0304].

Lee, as discussed above, discloses data transformation using a plurality of templates (templates are similar to transform definitions). However, Lee lacks the concurrent/parallel processing of multiple transform definitions.

However, Chung does (Col 7 ln 15-22; Col 7 ln 52-66). Chung discloses concurrently transforming data with multiple transformation definitions. The transform definitions are for example, aggregation transformation, filter transformation, rank transformation, etc.

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify Lee with Chung. One would have been motivated to do so in order to concurrently transform data so that "data contention is reduced and throughput is increased."

Claim 66: Lee as modified discloses the system of Claim 65 above, and further discloses means for selecting the transform process definition from a set

of transform process definitions, responsive to data associated with the data to be transformed [0308].

Claim 67: <u>Lee</u> as modified discloses the system of Claim 65 above, and further discloses wherein a second data element has no contribution to output data generated using the transform process definition, the second data element being a member of the plurality of data elements [0308].

Claim 68: <u>Lee</u> as modified discloses the system of Claim 65 above, and further discloses means for adding data to the output data file, the added data being configured responsive to the transform process definition and having no contribution from the data to be transformed [0308].

Conclusion

4.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of

time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire

THREE MONTHS from the mailing date of this action. In the event a first reply is

filed within TWO MONTHS of the mailing date of this final action and the advisory

action is not mailed until after the end of the THREE-MONTH shortened statutory

period, then the shortened statutory period will expire on the date the advisory

action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be

calculated from the mailing date of the advisory action. In no event, however, will

the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

Any inquiry concerning this communication or earlier communications from

the examiner should be directed to ALEX GOFMAN whose telephone number is

(571)270-1072. The examiner can normally be reached on Mon-Fri 9am-3pm

EST.

If attempts to reach the examiner by telephone are unsuccessful, the

examiner's supervisor, John Breene can be reached on (571)272-4107. The fax

phone number for the organization where this application or proceeding is

assigned is 571-273-8300.

Information regarding the status of an application may be obtained from

the Patent Application Information Retrieval (PAIR) system. Status information

for published applications may be obtained from either Private PAIR or Public

PAIR. Status information for unpublished applications is available through

Private PAIR only. For more information about the PAIR system, see http://pair-

direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-

free). If you would like assistance from a USPTO Customer Service

Representative or access to the automated information system, call 800-786-

9199 (IN USA OR CANADA) or 571-272-1000.

                                        Alex Gofman
                                        Examiner
                                        Art Unit 2162
3-16-10
AG

        /John Breene/
        Supervisory Patent Examiner, Art Unit 2162